

CS 216 Midterm 2

You MUST write your name and e-mail ID on EACH page and bubble in your userid at the bottom of EACH page – including this page.

If you do not do this, you will receive a zero for that page! (Or a grade penalty if you leave out the bubble form for this page)

If you are still writing when “pens down” is called, your exam will be ripped up and not graded – even if you are still writing to fill in the bubble forms. So please do that first. Sorry to have to be strict on this...

Other than bubbling in your userid at the bottom, please do not write in the footer section of each page.

There are 8 pages to this exam – once the exam starts, please make sure you have all 8 pages.

There are three types of questions: short, medium, and long answer. You can tell the difference because the short answer have a quarter page to answer, the medium have half a page to answer, and the long have a full page to answer the question. The short answer questions should not take more than a line or two to answer, and are worth 4 points each. *Your answer should not exceed about 20 words.* The medium answer questions are worth 6 points each, and the long answer questions are worth 12 points each. There are 100 points of questions and 105 minutes (1 hour 45 minutes) to take the exam, which means you should spend about 1 minute per question point; the extra 5 minutes are for bubbling in the exam footer.

This exam is CLOSED text book, closed-notes, **closed-calculator**, closed-cell phone, closed-computer, closed-neighbor, etc. Questions are worth different amounts, so be sure to look over all the questions and plan your time accordingly. Please sign the honor pledge here:

There are 10 types of people in the world – those that understand binary and those that don't.

(the bubble footer is automatically inserted in this space)

Arrays/Big-Oh

1. [4 points] What does big-Omega mean? Big-Theta?

2. [4 points] What does little-Oh mean? Little-Omega? Little-Theta?

3. [4 points] Given $f \in O(h)$ and $g \notin O(h)$ is the following statement true? Why or why not? For *all* positive integers $m, f(m) < g(m)$.

4. [4 points] Given $f \in O(h)$ and $g \notin O(h)$ is the following statement true? Why or why not? For *some* positive integers $m, f(m) < g(m)$.

(the bubble footer is automatically inserted in this space)

Trees, page 2

9. [12 points] Write an AVL tree that is in need of a double rotation to be properly balanced. Show the resulting tree both before and after said double rotation, as well as the intermediate step (i.e. between the two single rotations of the double rotation).

(the bubble footer is automatically inserted in this space)

Hashes, page 1

10. [4 points] What are the desired properties of a good hashing function?

11. [4 points] List two reasons why the size of a hash table should be a prime number.

12. [4 points] What special considerations have to be taken into account if you allow deletes from your hash table that uses a probing strategies for collision resolution?

13. [4 points] For each of the probing strategies for collision resolution (linear, quadratic, and double), where does the data structure try to fit the collided key?

(the bubble footer is automatically inserted in this space)

Hashes, page 2

14. [4 points] What problems are encountered with linear probing (ignoring deletes)?
15. [4 points] The Object class in Java contains a hashCode() method that returns an integer to be used as a key into a hash table (it must be modded by the hash table size, of course). The Object class does not know the type of the object that will call this (as it will be called by its subclasses). Keeping this in mind, how might this hash function work? We're looking for an English description here, not code.
16. [6 points] What is the running time for a hash table's operations (insert, find, printAll)? Why? If it differs for each of the collision resolution strategies, then explain for each strategy.
-

(the bubble footer is automatically inserted in this space)

IBCM, page 1 (and some C++)

17. [4 points] What limits exist with IBCM programming? In other words, what can you not write in IBCM? Why?
18. [4 points] What are the four instruction types in IBCM? Hint: the first type is the halt instruction. What are the other three?
19. [6 points] Given the C++ code: `int x[3][2] = { {1,2}, {3,4}, {5,6} }`, draw a diagram showing how this array is stored in memory.

(the bubble footer is automatically inserted in this space)

IBCM, page 2

20. [12 points] Write a **subroutine** in IBCM that will read in two values from the input, subtract them, and leave the result in the accumulator (we'll assume that the result should be first input number minus the second input number). Since this is a subroutine, it will be called with the BRL instruction, and you will need to return to the appropriate address. For extra credit, write it without using the SUB instruction (but make sure you get it right – if you don't get the rest of the subroutine correct, the extra credit won't help much). The IBCM opcodes are listed to the right.

IBCM	
0	halt
1	I/O
2	shifts
3	load
4	store
5	add
6	sub
7	and
8	or
9	xor
A	not
B	nop
C	jmp
D	jmpe
E	jmpl
F	brl

(the bubble footer is automatically inserted in this space)